putting

the globus toolkit®

www.globustoolkit.org

in its place:

the globus toolkit "ecosystem"
(and how to make it work for you)

lee liming
argonne national laboratory

the globus alliance

Getting Started

I. What problems is the Grid intended to address?
II. How far does the Globus Toolkit go toward solving these problems?
III. What other Grid software works well with the Globus Toolkit, and what can one do if those are added?

BREAK

IV. Planning Tasks
V. Roadmaps

what problems is the grid intended to address?

What Problems is the Grid Intended to Address?

The Grid is a highly pragmatic field.

- It arose from *applied* computer science.
- It is focused on *enabling* new types of applications.
- Funding and investment in the Grid has been motivated by the promise of *new capabilities*—not in computer science, but in other fields and in other areas of work.

What Kinds of Applications?

- Computation intensive
  - Interactive simulation (climate modeling)
  - Very large-scale simulation and analysis (galaxy formation, gravity waves, battlefield simulation)
  - Engineering (parameter studies, linked component models)
- Data intensive
  - Experimental data analysis (high-energy physics)
  - Image and sensor analysis (astronomy, climate study, ecology)
- Distributed collaboration
  - Online instrumentation (microscopes, x-ray devices, etc.)
  - Remote visualization (climate studies, biology)
  - Engineering (large-scale structural testing, chemical engineering)
- In all cases, the problems were big enough that they required people in several organization to collaborate and share computing resources, data, instruments.

What Types of Problems?

- Your system administrators can't agree on a uniform authentication system, but you have to allow your users to authenticate once (using a single password) then use services on all systems, with per-user accounting.
- You need to be able to offload work during peak times to systems at other companies, but the volume of work they'll accept changes from day-to-day.

What Types of Problems?

- You and your colleagues have 6000 datasets from the past 50 years of studies that you want to start sharing, but no one is willing to submit the data to a centrally-managed storage system or database.
- You need to run 24 experiments that each use six large-scale physical experimental facilities operating together in real time.

What Types of Problems?

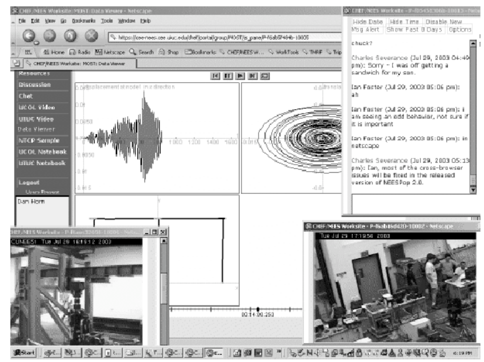
- Too hard to keep track of authentication data (ID/password) across institutions
- Too hard to monitor system and application status across institutions
- Too many ways to submit jobs
- Too many ways to store & access files and data
- Too many ways to keep track of data
- Too easy to leave "dangling" resources lying around (robustness)

Requirements "Themes"

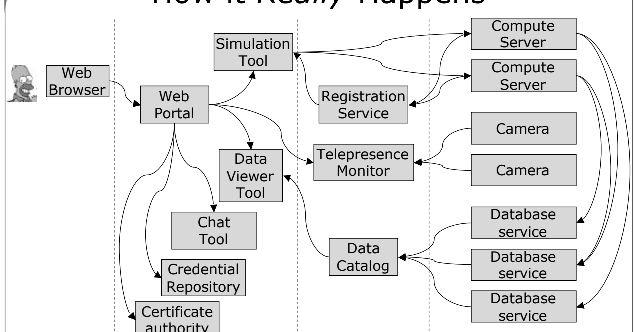
- Security
- Monitoring/Discovery
- Computing/Processing Power
- Moving and Managing Data
- Managing Systems
- System Packaging/Distribution

What End Users Need

Secure, reliable, on-demand access to data, software, people, and other resources (ideally all via a Web Browser!)



How it *Really* Happens



How it *Really* Happens

- Implementations are provided by a mix of
  - Application-specific code
  - "Off the shelf" tools and services
  - Tools and services from the Globus Toolkit
  - Tools and services from the Grid community (compatible with GT)
- Glued together by…
  - Application development
  - System integration

Why Does the Globus Toolkit Exist?

In the early 90s, Ian Foster (ANL, U-C) and Carl Kesselman (USC-ISI) enjoyed helping scientists apply distributed computing.

- Opportunities seemed ripe for the picking.
- *Application* of technology always uncovers new and interesting requirements.
- Science is cool!
- Big/Innovative science is even cooler!

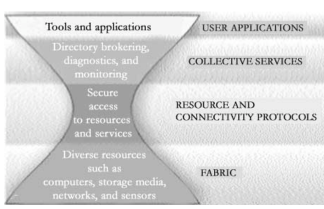
Why Does the Globus Toolkit Exist?

While helping to build/integrate a diverse range of applications, the same problems kept showing up over and over again.

- Too many different security systems
- Too many different scheduling/execution mechanisms
- Too many different storage systems
- Too many different monitoring/status/event systems

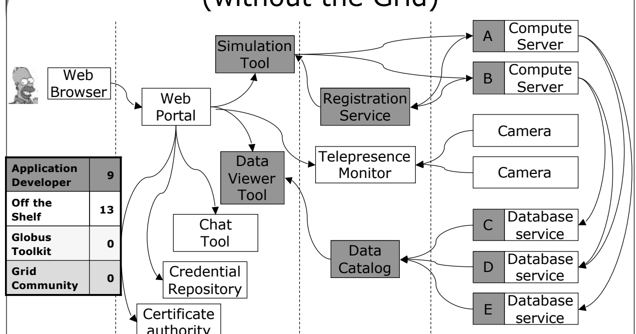
Forget Homogeneity!

- Trying to force homogeneity on users is futile. Everyone has their own preferences, sometimes even *dogma*.
- The Internet provides the model…



How it *Really* Happens (without the Grid)

Application Developer	9
Off the Shelf	13
Globus Toolkit	0
Grid Community	0



Users work with client applications

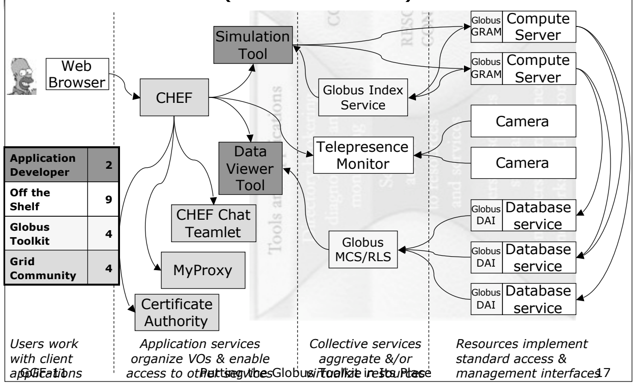
Application services organize VOs & enable access to other services

Collective services aggregate &/or virtualize resources

Resources implement standard access & management interfaces

the globus alliance
www.globus.org

How it *Really* Happens (with the Grid)



the globus alliance
www.globus.org

Who Is the Grid For?

- Any Grid (distributed/collaborative) application or system will involve several “classes” of people.
  - “End users” (e.g., Scientists, Engineers, Customers)
  - Application/Product Developers
  - System Administrators
  - System Architects and Integrators
- Each user class has unique skills and unique requirements.
- The user class whose needs are met varies from tool to tool (even within the Globus Toolkit).

the globus alliance
www.globus.org

Conclusions (so far)

1. Heterogeneity is great for administrators and overall system robustness (diversity), but *hard to account for in applications*.
2. “Durable” solutions (solutions that “stick”) address the needs of *all* user classes, not just some.

the globus alliance
www.globus.org

how far does the globus toolkit go toward solving these problems?

epcc KTH USC SCHOOL OF ENGINEERING

How Far Does the Globus Toolkit Go?

Executive Summary:

- The Globus Toolkit arose out of trying to solve real problems in real projects.
- It contains tools for addressing many challenging problems that come up in Grid applications/projects.
- These solutions make sense within a wide variety of applications.
- The Globus Toolkit does not solve *every* problem.
- The Globus Toolkit is not a "turn key" solution for *any* significant application or project.
- The Grid community has many tools that address other common challenges.
- "Turn key" solutions require integration.

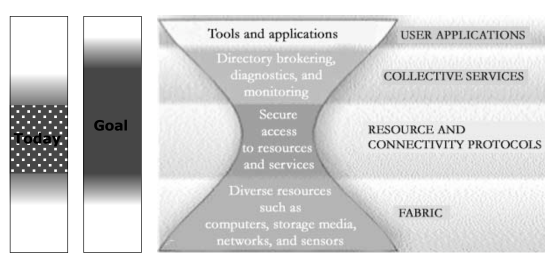
What Is the Globus Toolkit?

- The Globus Toolkit is a collection of solutions to problems that frequently come up when trying to build collaborative distributed applications.
- Heterogeneity
  - To date (v1.0 - v4.0), the Toolkit has focused on *simplifying heterogenity* for application developers.
  - We aspire to include more "vertical solutions" in future versions.
- Standards
  - Our goal has been to capitalize on and encourage use of existing standards (IETF, W3C, OASIS, GGF).
  - The Toolkit also includes reference implementations of new/proposed standards in these organizations.

"Standard Plumbing" for the Grid

- *Not* turnkey solutions, but *building blocks* and *tools* for application developers and system integrators.
  - Some components (e.g., file transfer) go farther than others (e.g., remote job submission) toward end-user relevance.
- Since these solutions exist and others are already using them (and they're free), it's easier to reuse than to reinvent.
  - And compatibility with other Grid systems comes for free!

How Far Does the Globus Toolkit Go?



Areas of Competence

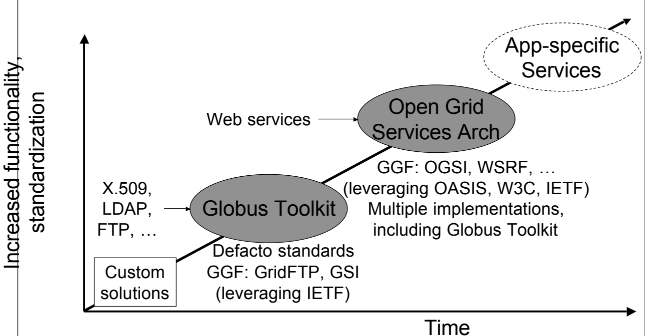
- "Connectivity Layer" Solutions
  - Service Management (WSRF)
  - Monitoring/Discovery (WSRF and MDS)
  - Security (GSI and WS-Security)
  - Communication (XIO)
- "Resource Layer" Solutions
  - Computing / Processing Power (GRAM)
  - Data Access/Movement (GridFTP, DAI)
  - In development: Telecontrol (NTCP/GTCP)
- "Collective Layer" Solutions
  - Data Management (RLS, MCS, DAI)
  - Monitoring/Discovery (MDS)
  - Security (CAS)

Leveraging Existing and Proposed Standards

- SSL/TLS v1 (from OpenSSL) (IETF)
- LDAP v3 (from OpenLDAP) (IETF)
- X.509 Proxy Certificates (IETF)
- GridFTP v1.0 (GGF)
- OGSI v1.0 (GGF)
- And others on the road to standardization:

  WSRF (GGF, OASIS), DAI, WS-Agreement, WSDL 2.0, WSDM, SAML, XACML

The "Grid Ecosystem"



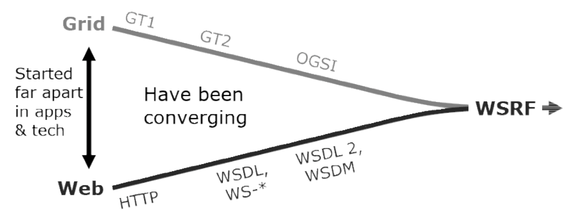
Putting it All Together: Open Grid Services Architecture

- Define a service-oriented architecture…
  - the key to effective virtualization
- …to address vital Grid requirements
  - AKA utility, on-demand, system management, collaborative computing, etc.
- …building on Web service standards.
  - extending those standards when needed

WSRF & WS-Notification

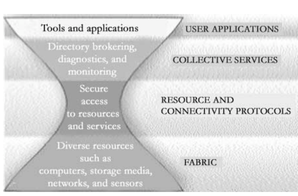
- Naming and bindings (basis for virtualization)
  - Every resource can be uniquely referenced, and has one or more associated services for interacting with it
- Lifecycle (basis for fault resilient state management)
  - Resources created by services following factory pattern
  - Resources destroyed immediately or scheduled
- Information model (basis for monitoring & discovery)
  - Resource properties associated with resources
  - Operations for querying and setting this info
  - Asynchronous notification of changes to properties
- Service Groups (basis for registries & collective svcs)
  - Group membership rules & membership management
- Base Fault type

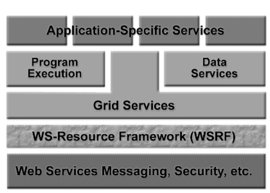
Grid and Web Services Convergence



The definition of WSRF means that the Grid and Web services communities can move forward on a common base.

Theory -> Practice

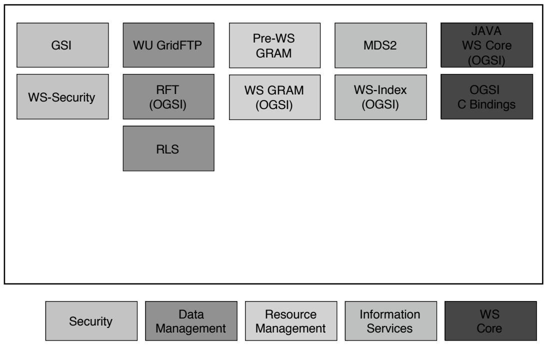




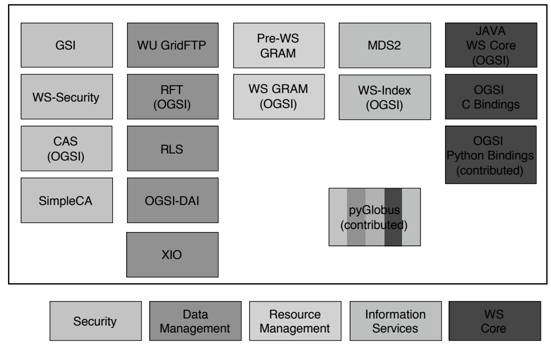
What You Get in the Globus Toolkit

- OGSI(3.x)/WSRF(4.x) Core Implementation
  - Used to develop and run OGSA-compliant Grid Services (Java, C/C++)
- Basic Grid Services
  - Popular among current Grid users, common interfaces to the most typical services; includes both OGSA and non-OGSA implementations
- Developer APIs
  - C/C++ libraries and Java classes for building Grid-aware applications and tools
- Tools and Examples
  - Useful tools and examples based on the developer APIs

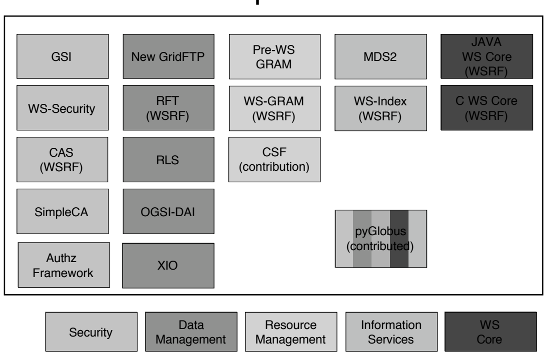
Components in Globus Toolkit 3.0



Components in Globus Toolkit 3.2



Planned Components in GT 4.0



Details – What Are You Getting Into?

- Installing the Globus Toolkit
- Configuring the Globus Toolkit
- Next Steps

Installation Overview

Prerequisite Software

Java SDK	C compiler
Apache Ant	GNU YACC (or Bison)
JUnit	GNU tar
Optional: Apache Tomcat	Optional: Microsoft .NET
Optional: JDBC-compliant DB	

Download Method

- Go to website, download a tar.gz file.
- Unzip/untar the file, run "install-gt3" script.
- Etc.

See http://www-unix.globus.org/toolkit/docs/3.2/ for details!

What to Download?

- Three choices for what to download
  - WS-Base Installer
    - Only the OGSI-compliant pieces; OGSI implementation plus OGSI-compliant services (GSI, GRAM, MDS, RFT, RLS, etc.)
    - Popular for new Grid applications & infrastructures; clearly the way to go if you're building something new.
  - Pre-WS Installer
    - Only the pre-OGSI pieces; pre-OGSI implementations of GSI, I/O, GRAM, MDS, GridFTP
    - Popular with people who've got existing Grid applications and infrastructure to support.
  - All Services
    - Everything above
    - Good for people who are doing new things but need compatibility with older stuff, too.
- Binary-only downloads for Linux (RedHat 7.3, RedHat 9, Fedora Core 1) and Solaris 9 are available.

See http://www-unix.globus.org/toolkit/docs/3.2/ for details!

Configuration Overview

- Set some environment variables
- Generate some certificates (do it yourself or obtain them from a certificate authority)
  - Host certificates (for servers)
  - User certificates (for users)
- Configure privileged scripts (run a script)
- Authorize specific users (edit a file)
- Test the installation (run a Grid Service client GUI)
- NOW, you can configure specific Grid Services… ☺

See http://www-unix.globus.org/toolkit/docs/3.2/ for details!

What Have You Got Now?

- A Grid development environment
  - Develop new OGSI-compliant Web Services
  - Develop applications using Grid APIs
- A set of basic Grid services
  - Job submission/management
  - File transfer (individual, queued)
  - Database access
  - Data management (replication, metadata)
  - Monitoring/Indexing system information
- Entry into Grid community software
  - Still more useful stuff!

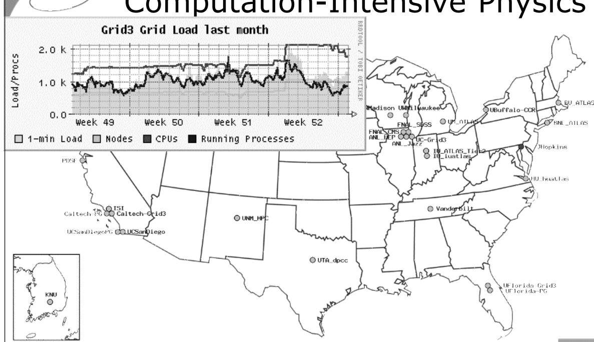
How To Use the Globus Toolkit

- By itself, the Toolkit has surprisingly limited *end user* value.
  - There's very little user interface material there.
  - You can't just give it to end users (scientists, engineers, marketing specialists) and tell them to do something useful!
- The Globus Toolkit is useful to *application developers* and *system integrators*.
  - You'll need to have a specific *application or system* in mind.
  - You'll need to have the right *expertise*.
  - You'll need to set up prerequisite *hardware/software*.
  - You'll need to have a *plan*.

Easy to Use – But Few Applications are "Easy"

- The uses that the Toolkit has been aimed at are *not* easy challenges!
- The Globus Toolkit makes them easier.
  - Providing solutions to the most common problems and promoting standard solutions
  - A well-designed implementation that allows many things to be built on it (lots of happy developers!)
  - 6+ years of providing support to Grid builders
  - Ever-improving documentation, installation, configuration, training

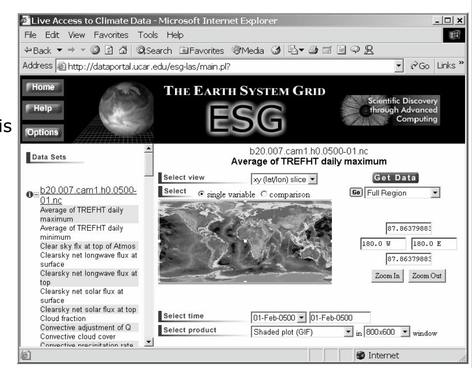
Grid2003: System for Computation-Intensive Physics



Status on 11/19/03 (http://www.ivdgl.org/grid2003)

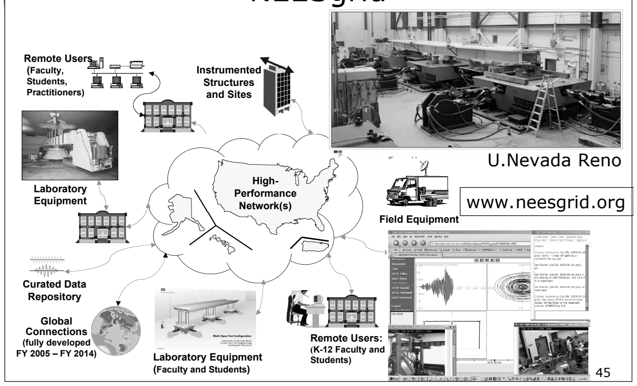
Earth System Grid

Goal: address technical obstacles to the sharing & analysis of high-volume data from advanced earth system models



the globus alliance
www.globus.org

Collaborative Engineering: NEESgrid



the globus alliance
www.globus.org

what other grid software works well with the globus toolkit, and what can one do if those are added?

the globus alliance
www.globus.org

The Globus Toolkit "Ecosystem"

- If the Globus Toolkit doesn't provide vertical solutions how have people achieved things with the Grid?
- What else is out there and how does the Globus Toolkit fit with it all?
- What others tools and technologies should we be looking at when building Grid systems/applications/products?

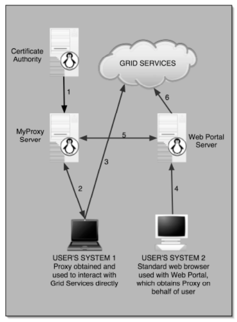
the globus alliance
www.globus.org

Security Tools

- Certificate Management
  - Getting users "signed up" to use the Grid
  - Getting the user's Grid credentials to wherever they're needed in the system
- Authorization/Access Control
  - Tools for storing and providing access to system-wide authorization information
  - Central data store for supporting decentralized control mechanisms

MyProxy

- MyProxy is a remote service that stores user credentials.
  - Users can request proxies for local use on any system on the network.
  - Web Portals can request user proxies for use with back-end Grid services.
- Grid administrators can pre-load credentials in the server for users to retrieve when needed.
- Greatly simplifies certificate management!



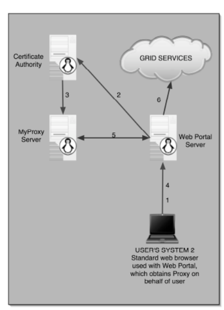
KX.509 and KCA

- Institutions that already have a Kerberos realm can use KX.509 and KCA to provide local users with Grid proxy certificates without using a Certificate Authority.
- When users authenticate with Kerberos, they may obtain proxy certificates in addition to their Kerberos tickets.
- KCA is a Kerberized certification service, and KX.509 is a Kerberized client that generates and stores proxy certificates.
- Unlike MyProxy, KX.509 and KCA create credentials for users, so remote sites must be configured to trust the local KCA service's certification authority.

PKINIT

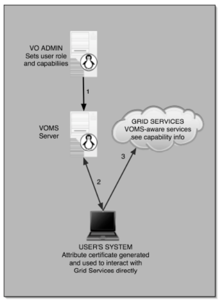
- PKINIT is a service that allows users to use Grid certificates to authenticate to a Kerberos realm.
- For sites that use Kerberized services (like AFS), this allows remote Grid users to obtain the necessary Kerberos tickets to use the site's local facilities properly.
- PKINIT replaces the Kerberos "klog" command and uses the user's Grid certificate to eliminate the need for a Kerberos passphrase.

Earth System Grid User Registration Service



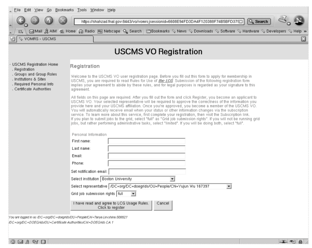
- Portal extensions (CGI scripts) that automate user registration requests.
  - Solicits basic data from user.
  - Generates cert request from ESG CA (implemented with "simple CA" from GT).
  - Admin interface allows CA admin to accept/reject request.
  - Generates a certificate and stores in MyProxy service.
  - Gives user ID/password for MyProxy.
- Benefits
  - Users never have to deal with certificates.
  - Portal can get user cert from MyProxy when needed.
  - Database is populated with user data.
- This can be reused in other projects!

VOMS



- An alternative to GT CAS
- Database of user roles and capabilities
  - Administrative tools
  - Client interface
- voms-proxy-init
  - Uses client interface to produce an attribute certificate (instead of proxy) that includes roles & capabilities signed by VOMS server
  - Works with non-VOMS services, but gives more info to VOMS-aware services
- Allows VOs to centrally manage user roles and capabilities

VOX and VOMRS



Extends VOMS to include an ESG-like registration service

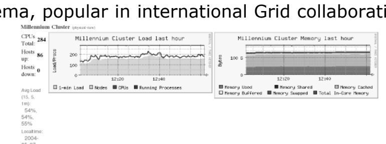
- Web registration interface
- Builds user database with extended fields
- Populates VOMS server

Monitoring/Discovery Tools

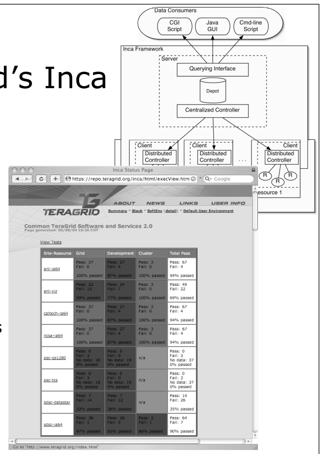
- Specialized Information Providers
  - "Hookups" to sources of information
  - Handle "last mile" integration with specialized equipment
- Alternative Aggregators
  - Alternate mechanisms for storing and provide access to system information
- Visualizers and User Interfaces
  - Viewing and display tools for showing system information for a variety of specialized purposes

Ganglia Cluster Toolkit

- Ganglia is a toolkit for monitoring clusters and aggregations of clusters (hierarchically).
- Ganglia collects system status information and makes it available via a web interface.
- Ganglia status can be subscribed to and aggregated across multiple systems.
- Integrating Ganglia with MDS services results in status information provided in the proposed standard GLUE schema, popular in international Grid collaborations.

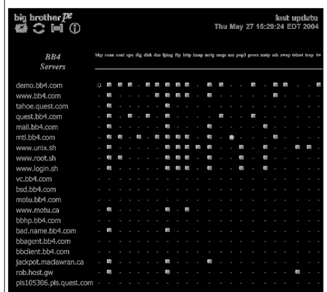


TeraGrid’s Inca



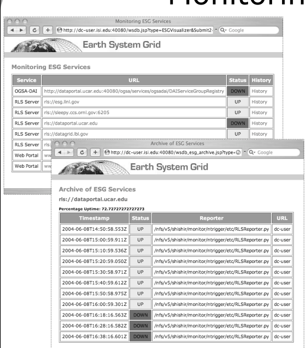
- Hierarchical Status Monitoring
  - Groups tests into logical sets
  - Supports many levels of detail and summarization
- Flexible, scalable architecture
  - Very simple reporter API
  - Can use existing test scripts (unit tests, status tools)
  - Hierarchical controllers
  - Several query/display tools

Big Brother



- Big Brother provides a scripting framework for producing web-based system status displays.
- Arbitrary scripts can be “plugged in” and executed across many systems, with results displayed on a single page.
- BB can serve as a nice front-end to MDS or other monitoring systems.

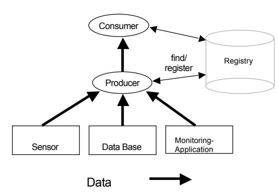
Earth System Grid Monitoring Services



- Used to monitor current status and history of ESG RLS, OGSA-DAI, and Web Portal services.
- Based on GT Index Service, with additions:
  - Archiver Service
  - Trigger Service
  - Web-based Service Data Viewer
  - Email alerts
  - Python test scripts
- Everything can be reused!

R-GMA

- An alternative to GT Indexing Services
  - Relational Database approach
  - Based closely on GGF Grid Monitoring Architecture (GMA)
  - Can be used with GT Pre-WS information providers
- Producer/Consumer architecture
  - Matchmaking
  - Agents
  - Static & Stream data
  - Supports hierarchy
- Used on most EU Grids



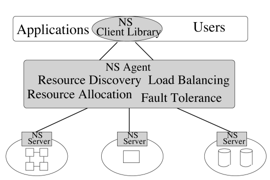
Computing/Processing Tools

- Workflow Managers
  - Organize and coordinate task execution within a complicated application
  - Often coordinates data movement and task execution
- Metaschedulers
  - Optimize use of distributed compute pools
- Virtual Data Tools
  - Manage the trade-off between data storage and processing power

Condor, Condor-G, DAGman

- Condor addresses many workflow challenges for Grid applications.
  - Managing sets of subtasks
  - Getting the tasks done reliably and efficiently
  - Managing computational resources
- Similar to a distributed batch processing system, but with some interesting twists.
  - Scheduling policy
  - ClassAds
  - DAGman
  - Checkpointing and Migration
  - Grid-aware & Grid-enabled
  - Flocking (linking pools of resources) & Glide-ins

NetSolve/GridSolve



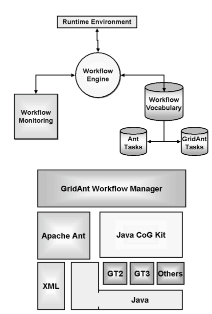
An RPC-based library for executing solver code on Grid resources

- Client/Agent/Server architecture
- Very simple client model
- Integrated with a wide variety of development environments (MATLAB, Octave, C, FORTRAN, etc.)
- Supports many types of compute resources and authentication systems
- Server runs in user space
- Agent handles Grid complexity

GridAnt

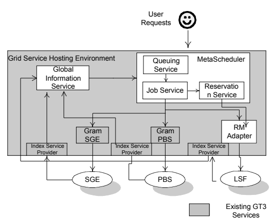
A user-controllable, grid-enabled workflow engine

- Uses Java Ant for workflow execution
- Uses Java CoG and GT services to interface to system services



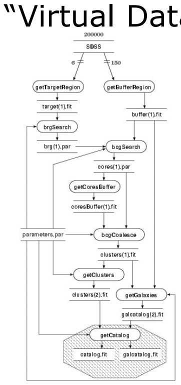
Platform CSF

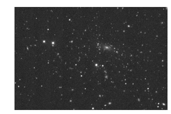
- An open source implemenation of OGSA-based metascheduler for VOs.
  - Supports emerging WS-Agreement spec
  - Supports GT GRAM
  - Uses GT Index Service
- Fills in gaps in existing resource management picture
  - Integrated with Platform LSF and Platform Multicluster
  - Anticipated for inclusion in GT 4.0 release



Chimera "Virtual Data"

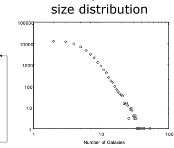
- Captures both logical and physical steps in a data analysis process.
  - Transformations (logical)
  - Derivations (physical)
- Builds a catalog.
- Results can be used to "replay" analysis.
  - Generation of DAG (via Pegasus)
  - Execution on Grid
- Catalog allows introspection of analysis process.





Sloan Survey Data

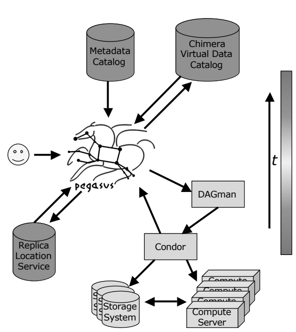
Galaxy cluster size distribution



Pegasus Workflow Transformation

Converts Abstract Workflow (AW) into Concrete Workflow (CW).

- Uses Metadata to convert user request to logical data sources
- Obtains AW from Chimera
- Uses replication data to locate physical files
- Delivers CW to DAGman
- Executes using Condor
- Publishes new replication and derivation data in RLS and Chimera (optional)



Data Tools

- Virtual Data Tools
  - Manage the trade-off between data storage and processing power (already covered)
- Movement/Transfer Tools
  - Interfaces that meet specialized application or user needs
  - "Last mile" integration to specialized storage systems
- Optimization Tools
  - Help optimize the use of storage systems for specialized user communities

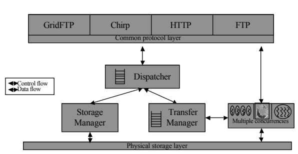
GSI-SCP/SFTP

- GSI-OpenSSH is a version of OpenSSH that supports Grid authentication.
  - Remote terminal (gsi-ssh)
  - Remote Copy (gsi-scp)
  - Secure FTP (gsi-sftp)
- More familiar to many users than GridFTP.
- Doesn’t take advantage of GridFTP’s capabilities (parallelism, partial files, third-party transfers, etc.)

UberFTP

- UberFTP is an interactive (text prompt) client for GridFTP.
- Supports more features than NCFTP
  - Parallelism
  - Third-party transfer

NeST



- Grid-optimized Storage Appliance
  - Multiprotocol (pluggable) access mechanisms
  - Pluggable storage system layer (flexible)
  - Supports reservations
  - Issues ClassAds (for use with Condor matchmaking)
  - Can be run at user level
  - Can manage users/groups
  - “No Futz” installation
- Supports creation of I/O communities
  - Combine the jobs-to-data and data-to-jobs strategies
  - Users can build their own NeSTs
  - Works very well with Condor and Condor applications

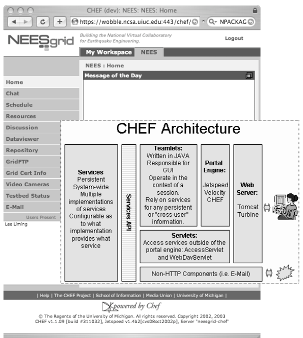
Collaboration Tools

- Web Portals
  - Tools for building web interfaces that provide access to system/application capabilities
- High-function Virtual Meeting Spaces
  - Tools for facilitating collaborative work in realtime with lots of gadgets

the globus alliance
www.globus.org

Sakai

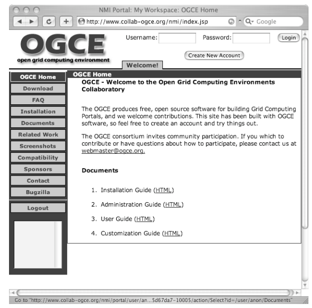
CHEF/Sakai



- The **C**ompre**H**ensive collaborativ**E** **F**ramework (**CHEF**) is a flexible environment for supporting distributed learning and collaborative work.
- CHEF is rapidly evolving into Sakai, with emphasis on JSR-168 and localization.
- CHEF is highly extensible with support for JetSpeed, Velocity, and other portal interfaces.

the globus alliance
www.globus.org

Open Grid Computing Environment (OGCE)



- Extends CHEF/Sakai to include support for Grid services
  - MyProxy
  - GridPort
  - GT services (GRAM, GridFTP, MDS, etc.)
  - Java CoG
- Provides a "quick start" for building Grid-enabled portals.

the globus alliance
www.globus.org

Access Grid

- Provides seamless group-to-group collaboration spaces for groups that are not co-located
  - Immersive audio (sounds like "everyone is here")
  - Multiple multicast video streams (multipoint "everyone sees everyone")
  - Can integrate other video sources into the space
  - Display walls common, but not required
- "Virtual spaces" (like channels) allow people to find each other
- Ideally, groups can work together without thinking about the technology.
- Increasingly Grid-enabled

the globus alliance
www.globus.org

System Packaging/Distribution

- Distribution and Packaging Tools
  - Getting software distributed and installed uniformly throughout a broad collaboration
  - Tools that help create integrated distributions that work on a wide variety of systems
- Integrated Distributions
  - Customized distributions of common Grid software

PACMAN

- PACMAN is a package manager, facilitating obtaining, installing, and updating software distributions.
  - "Caching" paradigm allows distributors to provide distributions to users.
  - Caches can include configuration information, aiding in maintaining common configuration settings.
  - Caches allow users to easily get the latest software for a distribution.
  - Largely agnostic about other packaging mechanisms (tar.gz, GPT, RPM).
- PACMAN is used in virtual organizations to maintain a common software base across institutions & sites.

Grid Packaging Tools (GPT)

- GPT is the packaging used for the Globus Toolkit, but it exists independently.
  - Adds metadata to tar.gz files, putting more "intelligence" into build/install/config
  - Tools for developers and users
- Focus is multiplatform, tricky builds
  - Works on most Unix systems
  - Source & Binary packages
  - Dependency management
  - Relocatable installations (multiple installs)
  - Setup (config) awareness
  - Bundles (aggregations of packages)

NSF Middleware Initiative (NMI)

- The NSF-sponsored NMI effort produces a value-added collection of Grid software.
  - Binary distributions for popular platforms
  - Integration points
  - Enhanced testing
  - Documentation
  - Support
- Covers a lot of useful software
  - APST, Condor, CPM, DataCutter, DataCutter STORM, Globus Toolkit, GPT, Gridconfig, GridPort, GridSolve, GSI OpenSSH, Inca, KX.509/KCA, Look, MPICH-G2, MyProxy, Network Weather Service, OpenSAML, PERMIS, PyGlobus, Shibboleth, SRB Client, UberFTP, WebISO (Web Initial Sign-on)

NPACKage

- Custom distribution created for NPACI sites.
- Includes all NMI components, plus…
  - SRB
  - DataCutter
  - Ganglia
  - APST
  - LaPACK for Clusters (LFC)
  - Network Weather Service
- Uses PACMAN for distribution and installation.

NPACI Rocks

- Rocks is primarily a software distribution for Linux clusters.
  - OS, common tools, cluster-specific tools
  - Very easy to install, requires very little systems expertise to get up & running
  - Homogenous (for those who like that)
  - Extras can be added as "rolls"
- Includes a Grid Roll (not to be confused with the Grid Engine Roll)
  - All of the NMI software (a very easy way to get this stuff installed on your cluster!)

Virtual Data Toolkit (VDT)

- VDT is a grid middleware distribution focused on the needs of the NSF-funded GriPhyN and iVDGL projects, both of which are focused on Physics and Astronomy applications.
  - Ease of use (and installation) is key.
- Contents
  - Globus Toolkit & Condor, Condor-G
  - Virtual Data Tools (Chimera, Pegasus, RLS)
  - Utilities (GSI-OpenSSH, UberFTP, MonaLisa, MyProxy, KX.509, etc.)
- Uses PACMAN for distribution, install, configuration.
- Deployed on Grid3 (28 major U.S. sites)

break

(next up)
planning tasks
roadmaps

planning tasks

the globus alliance
www.globus.org

Review: How it *Really* Happens

- Implementations are provided by a mix of
  - Application-specific code
  - “Off the shelf” tools and services
  - Tools and services from the Globus Toolkit
  - Tools and services from the Grid community (compatible with GT)
- Glued together by…
  - Application development
  - System integration

the globus alliance
www.globus.org

Planning Tasks

1. Identify project/application goals
2. Identify specific functional requirements
3. Identify/develop social policies, procedures, processes
4. Identify components to meet requirements
5. Form an integration plan and identify functional gaps
6. Implement the product
7. Deploy the product
8. Provide O&M for the product, assess status and iterate

the globus alliance
www.globus.org

Application “Vision”

- Never underestimate the amount of work that can be wasted for lack of a specific target.
  - Desired end user capabilities (accomplishments?)
  - …leads to: specific end user goals, milestones
  - …leads to: specific system *requirements*
  - …leads to: specific system *features*
  - …leads to: specific system *components*
- A specific target application/system will focus the work dramatically. For example:
  - Enable engineers to run *many* more simulation runs.
  - Enable realtime remote observation of MRI scanning.
  - Provide broad, easy-to-use access to satellite data.
  - Encourage scientists to share/use experimental data.

the globus alliance
www.globus.org

Functional Requirements

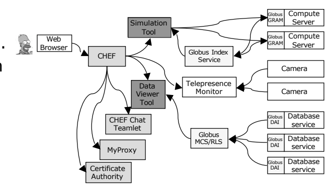
- What capabilities must the system or application provide?
  - *“What?”* Not *“How?”*
  - A common mistake is to select components before defining what is really needed.
  - Requirements should clearly relate to project goals.
- Examples:
  - Experiment participants must be able to communicate in realtime with each other.
  - The system must be able to manage and carry out a large number of inter-related tasks efficiently.
  - Detailed records of how data was derived must be kept and made available to data users.
  - Data must be automatically replicated among servers to optimize proximity based on past use.
  - Access control must be employed across the system, with end-user “single sign-on.”

Social Policies/Procedures

- How will people use the system?
  - Who will set up access control?
  - Who creates the data?
  - How will computational resources be added to the system?
  - How will simulation capabilities be used?
  - What will accounting data be used for?
- Not all problems are solved by technology!
- Understanding how the system will be used is important for narrowing the requirements.

Architecture

- Once you have some decent requirements and some understanding of use cases…
  - Draw the system design.
  - Describe how the design will meet the needs of typical use cases.
  - Consider deployment and M&O requirements for the design.
  - Get feedback!
- You will start getting a sense of what components will be needed.



Select Components

- Within the system design, components will have functional requirements, too.
  - Capabilities ("features")
  - Interfaces (protocols, APIs, schema)
  - Performance/scalability metrics
- Ideally, much of it already exists.
  - Leverage what's already out there (Web, Grid, fabric technologies, off-the-shelf products, etc.).
  - Decompose into smaller bits if necessary.
  - If too much is unique to this application, you're probably doing something wrong.
  - If a candidate component is almost--but not quite--perfect, it can probably be extended (or used in conjunction with something else) to meet requirements.

Integration Plan

- Existing components must be integrated.
  - Identify "integration points"
  - Define interfaces
  - Develop "glue" if necessary
- New components must be developed.
  - Identify requirements (features+interfaces+performance)
  - Plan development

Application Development

- Phased "top-down" development
  - Focus on satisfying individual project goals or requirements in turn, or
  - Focus on widening deployment in turn.
  - Danger of "muddying" the architecture (inefficiencies creep in, especially regarding reusability).
- "Bottom-up" development
  - Focus first on components, then move to "system integration".
  - Danger of missing the "big picture" (missing unstated requirements).

Deployment

- Involve "real users" as early as possible.
  - You'll learn a lot and be able to "course correct."
  - You'll establish "happy users" to help in later stages.
- Pick early adopters carefully.
  - Aggressive users, technologically skilled, representative of the target user base.
  - Set expectations carefully.
  - Be wary of overinvestment.
- Deployment is a significant chunk of your effort.
  - Separate team?
  - Make sure it's linked to the development activity.

Operations, Evaluation, Iteration

- O&M on Grids can be significant.
  - "Virtual organization" data management
  - System monitoring & troubleshooting
  - Keeping up with evolving technologies
  - Validation of new deployments
- Constant evaluation is important.
- Be prepared to start all over again with increased experience.
  - Refinement, reengineering
  - Extending into new areas

Expertise

- The following types of skills are needed.
  - System Administration
  - System Integration
  - Application Development
  - Planning/Management
- You need less of these with the Toolkit than without it.
  - Existing services can be re-used.
  - APIs make development of new services easier.
- In most projects, individuals will combine skills above. (You don't need one person for each.)

Prerequisite Hardware/Software

- Most Grids make heavy use of Unix/Linux
  - Open Source and easily extensible
  - Decent price/performance/reliability/native security
  - More familiar to original Grid community
- Windows .NET is good for OGSI/WSRF-based work
  - Yet another benefit of being Web service standards compatible
- Hardware depends on application goals
  - Clusters for compute-intensive apps, highly-parallel work (including data striping) and databases
  - Mass storage for data-intensive apps
  - Networking for communication-intensive apps (data transfer, videoconferencing, remote visualization, etc.)

Important Planning Considerations

- All Grid technology is evolving rapidly.
  - Web services standards
  - Grid interfaces
  - Grid implementations
  - Grid hosting services (ASP, SSP, etc.)
- Community is important!
  - Best practices (GGF, OASIS, etc.)
  - Open source (Linux, Axis, Globus, etc.)
- Application of community standards is *vital*.
  - Increases leverage
  - Mitigates (a bit) effects of rapid evolution
  - Paves the way for future integration/partnership

roadmaps

Roadmaps

- Grid2003 - Computation Intensive
- Earth System Grid - Data Intensive
- NEESgrid - Distributed Collaboration

Computation-Intensive Science: Grid2003

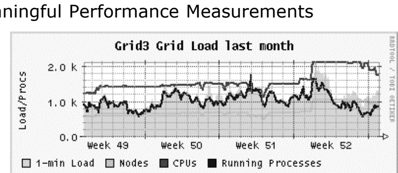
- GriPhyN - Grid Physics Network (NSF)
- iVDGL - International Virtual Data Grid Laboratory (NSF)
- LCG - LHC Computing Grid (EU)
- PPDG - Particle Physics Data Grid (DOE)

Grid2003 Project Goals

- Ramp up U.S. Grid capabilities in anticipation of LHC experiment needs in 2005.
  - Build, deploy, and operate a working Grid.
  - Include all U.S. LHC institutions.
  - Run real scientific applications on the Grid.
  - Provide state-of-the-art monitoring services.
  - Cover non-technical issues (e.g., SLAs) as well as technical ones.
- Unite the U.S. CS and Physics projects that are aimed at support for LHC.
  - Common infrastructure
  - Joint (collaborative) work

Grid2003 Requirements

- General Infrastructure
- Support Multiple Virtual Organizations
- Production Infrastructure
- Standard Grid Services
- Interoperability with European LHC Sites
- Easily Deployable
- Meaningful Performance Measurements



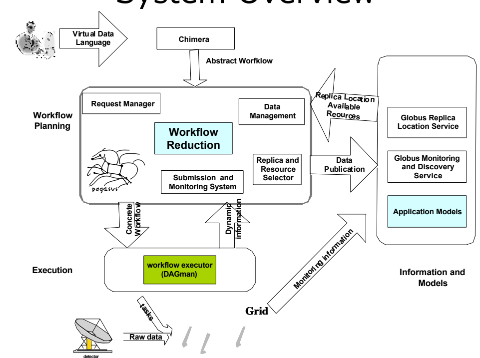
Grid2003 Components

- GT GRAM
- GT MDS
- GT GridFTP
- GT RLS
- GT MCS
- Condor-G
- DAGman
- Chimera & Pegasus
- GSI-OpenSSH
- MonALISA
- Ganglia
- VOMS
- PACMAN

Grid2003 Components

- **Computers & storage** at 28 sites (to date)
  - 2800+ CPUs
- **Uniform service environment** at each site
  - Globus Toolkit provides basic authentication, execution management, data movement
  - Pacman installation system enables installation of numerous other VDT and application services
- **Global & virtual organization services**
  - Certification & registration authorities, VO membership services, monitoring services
- **Client-side tools** for data access & analysis
  - Virtual data, execution planning, DAG management, execution management, monitoring
- **IGOC**: iVDGL Grid Operations Center

System Overview



Grid2003 Operation

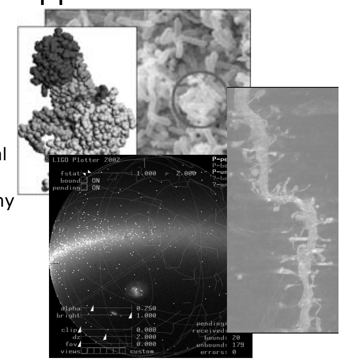
- All software to be deployed is integrated in the Virtual Data Toolkit (VDT) distribution.
  - The VDT uses PACMAN to ease deployment and configuration.
  - Each participating institution deploys the VDT on their systems, which provides a standard set of software and configuration.
  - A core software team (GriPhyN, iVDGL) is responsible for VDT integration and development.
- A set of centralized services (e.g., directory services) is maintained Grid-wide.
- Applications are developed with VDT capabilities, architecture, and services directly in mind.

Grid2003 Deployment

- VDT installed at more than 25 U.S. LHC institutions, plus one Korean site.
- More than 2000 CPUs in total.
- More than 100 individuals authorized to use the Grid.
- Peak throughput of 500-900 jobs running concurrently, completion efficiency of 75%.

Grid2003 Applications

- 6 VOs, 11 Apps
- High-energy physics simulation and data analysis
- Cosmology based on analysis of astronomical survey data
- Molecular crystalography from analysis of X-ray diffraction data
- Genome analysis
- System “exercising” applications



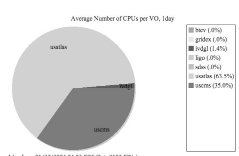
Grid2003 Applications To Date

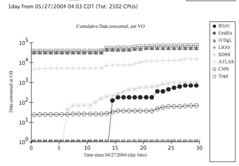
- CMS proton-proton collision simulation
- ATLAS proton-proton collision simulation
- LIGO gravitational wave search
- SDSS galaxy cluster detection
- ATLAS interactive analysis
- BTeV proton-antiproton collision simulation
- SnB biomolecular analysis
- GADU/Gnare genone analysis
- Various computer science experiments

www.ivdgl.org/grid2003/applications

Grid2003 Interesting Points

- Each virtual organization includes its own set of system resources (compute nodes, storage, etc.) and people. VO membership info is managed system-wide, but policies are enforced at each site.
- *Throughput* is a key metric for success, and monitoring tools are used to measure it and generate reports for each VO.

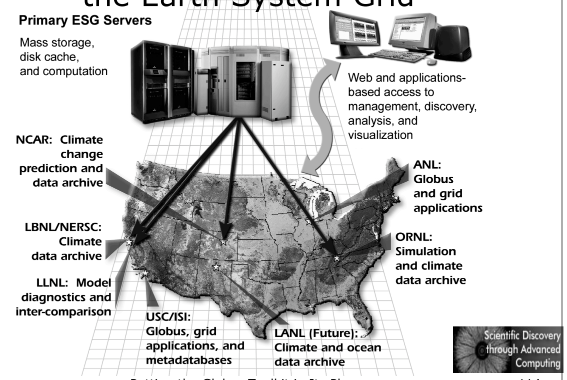




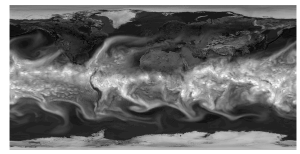
Grid2003 Metrics

Metric	Target	Achieved
Number of CPUs	400	2762 (28 sites)
Number of users	> 10	102 (16)
Number of applications	> 4	10 (+CS)
Number of sites running concurrent apps	> 10	17
Peak number of concurrent jobs	1000	1100
Data transfer per day	> 2-3 TB	4.4 TB max

Data-Intensive Science: the Earth System Grid



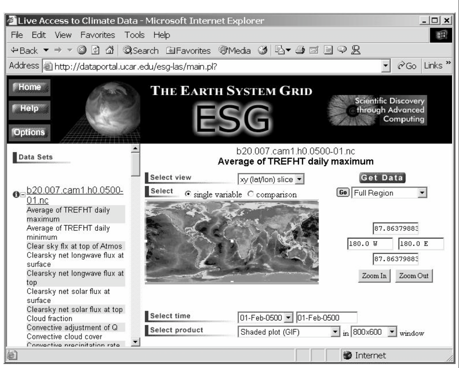
ESG Project Goals



- Improve productivity/capability for the simulation and data management team (data producers).
- Improve productivity/capability for the research community in analyzing and visualizing results (data consumers).
- Enable broad multidisciplinary communities to access simulation results (end users).
- The community needs an integrated "cyberinfrastructure" to enable smooth *workflow* for *knowledge development*: compute platforms, collaboration & collaboratories, data management, access, distribution, and analysis.

Earth System Grid

Goal: address technical obstacles to the sharing & analysis of high-volume data from advanced earth system models



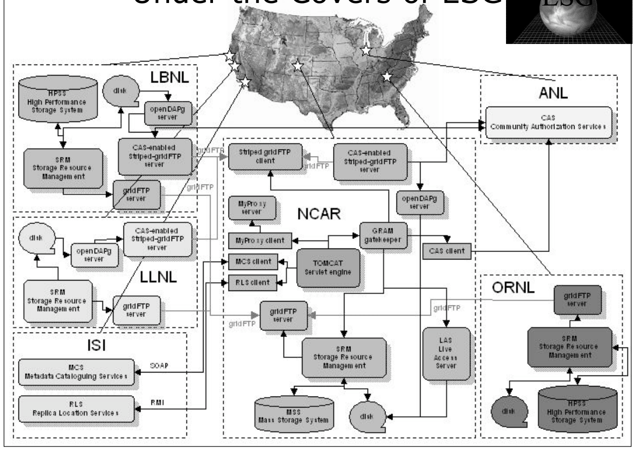
ESG Requirements

- Move data a minimal amount, keep it close to computational point of origin when possible.
- When we must move data, do it fast and with a minimum amount of human intervention.
- Keep track of what we have, particularly what's on deep storage.
- Make use of the facilities available at a number of sites. (Centralization is not an option.)
- Data must be easy to find and access using standard Web browsers.

Major ESG Components

- Grid Services
  - GRAM
  - GridFTP (+striped GridFTP server)
  - MDS (+WebSDV, +Trigger Service, +Archiver)
  - MyProxy
  - SimpleCA
  - RLS
  - MCS
- Other Services
  - OpenDAPg
  - HPSS
  - SRM
  - Apache, Tomcat
- ESG-specific services
  - Workflow Manager
  - Registration Service

Under the Covers of ESG



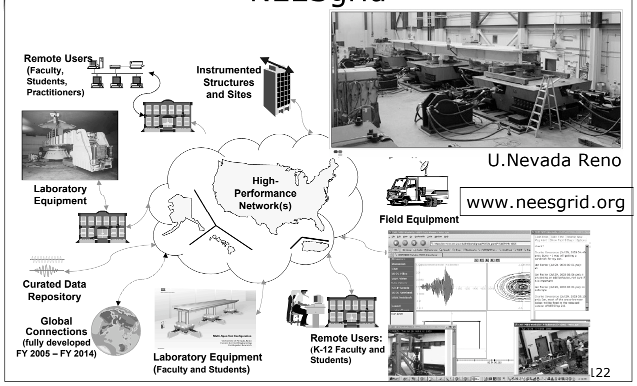
ESG Deployment

- Four data centers (LBNL, LLNL, NCAR, ORNL)
- User registration and authorization established
- Two major datasets are available, with associated metadata
- Work underway to add IPCC datasets as they are produced

ESG Interesting Points

- A lot of effort has been needed to build acceptable metadata models.
- Ease of use (simple interfaces, like registration service) is critical!
  - Users shouldn't have to see anything other than web interface and the data they ask for.
  - Don't bother giving certificates to users as long as they're using the portal for everything.
- Specific goals (e.g., providing access to specific datasets) will dramatically focus work.

Collaborative Engineering: NEESgrid



NEESgrid System Integrators

- National Center for Supercomputing Applications (NCSA)
- Argonne National Laboratory
- USC-Information Sciences Institute
- University of Michigan
- Stanford University
- UC-Berkeley
- Pacific Northwest National Laboratory

NSF's Goals for NEESgrid

- Encourage collaboration among earthquake engineering researchers and practitioners.
  - Provide remote access to large-scale NSF earthquake engineering facilities.
  - Provide distributed collaboration tools.
  - Provide easy-to-use simulation capabilities.
  - Allow integration of physical and simulation capabilities.
  - Provide a community data repository for sharing data generated by use of the system.
- Create a *cyberinfrastructure* for earthquake engineering.
  - Define and implement Grid-based integration points for system components.

NEESgrid Core Capabilities

- Tele-control and tele-observation of experiments
- Data cataloging and sharing
- Remote collaboration and visualization tools and services
- Simulation execution and integration

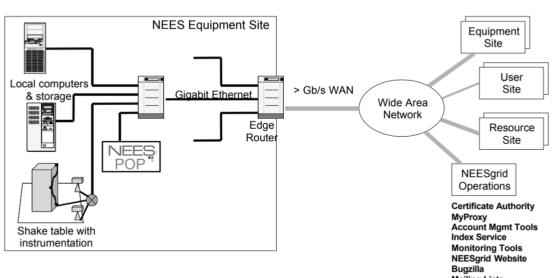
NEESgrid Requirements

- Single sign-on with Grid credentials
- Web interfaces for end users
  - Collaboration services (chat, video, documents, calendars, notebooks, etc.)
  - Telepresence services (video feeds)
  - Telecontrol (in limited instances)
  - Data viewing, data browsing and searching
  - Simulation capabilities
- Uniform interfaces for major system capabilities
  - Control
  - Data acquisition
  - Data streams
  - Data repository services

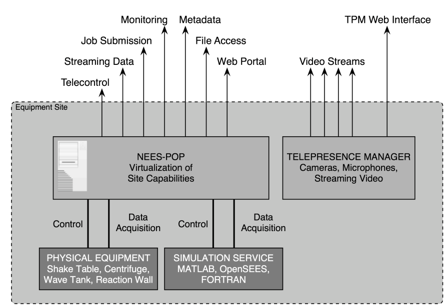
More NEESgrid Requirements

- System security
  - Protect facilities from misuse
  - Physical safety!
- Distributed collaboration during realtime experiments
- Automated (pre-programmed) control of distributed experiments (physical *and* simulation)
- Simplify effects of heterogeneity at facilities

NEESgrid High-level Structure



Architecture of NEESgrid Equipment Site



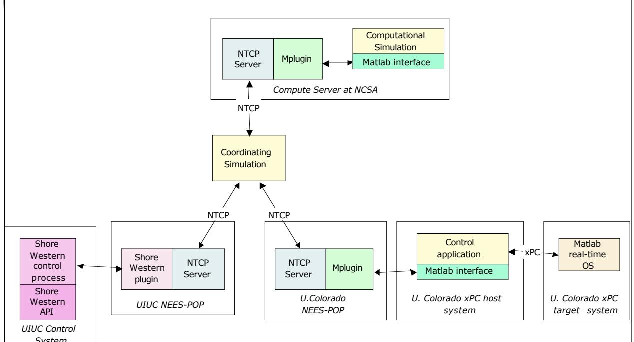
Major NEESgrid Components

- OGSA Services
  - NTCP - Uniform Telecontrol Interface
  - NMDS - Metadata Repository Management
  - NFMS - File Repository Management
- Creare Data Turbine - Data & Video
- CHEF - Web Portal, Collaboration Tools
- NEESgrid Simulation Portal - Simulation Tools
- OpenSEES, FedeasLab - Simulation Frameworks
- Other Grid Services
  - MyProxy - Authentication
  - GridFTP - File Movement
  - GRAM - Job Submission/Management
  - MDS, Big Brother - System Monitoring
  - GSI-OpenSSH - Administrative Logins
  - GPT - Software Packaging

Tele-Control Services

- Transaction-based protocol and service (NTCP) to control physical experiments and computational simulations.
- OGSI-based implementation (GT3.2)
- Plug-ins to interface the NTCP service
  - A computational simulation written in Matlab
  - Shore Western control hardware
  - MTS control hardware (via Matlab and xPC)
  - LabView control software
  - Still-image camera control
  - DAQ triggering
- Security architecture, including GSI authentication and a flexible, plug-in-based authorization model.

Tele-Control Services



NEESgrid Deployment

- NEES-POPs installed at 16 facilities
- Experiment-based Deployment (EBD)
  - Sites propose experiments
  - SI and sites cooperatively run experiment using NEESgrid (deployment)
  - Tests architecture and components, identifying new requirements
- October 2004 transition to M&O team (SDSC)
- First round of research proposals also begin in October 2004
- Grand Opening in November 2004

NEESgrid Interesting Points

- Requirements are hard to define when a community is unused to collaboration.
  - Early deployment and genuine use is critical for focusing work.
  - Iterative design is useful in this situation.
- Considerable effort has been needed for data modeling (still unproven).
- Plug-in interfaces ("drivers") are much more useful than originally imagined.
- Real users don't want to deal with WSDL. They need user-level APIs.

conclusions

Lessons Learned

- The Globus Toolkit has useful stuff in it.
- To do anything significant, a lot more is needed.
  - The Grid community (collectively) has many useful tools that can be reused!
  - System integration expertise is mandatory.
- OGSA and community standards (GGF, OASIS, W3C, IETF) are extremely important in getting all of this to work together.
- There's much more to be done!

"Standard Plumbing" for the Grid

- *Not* turnkey solutions, but *building blocks* and *tools* for application developers and system integrators.
  - Some components (e.g., file transfer) go farther than others (e.g., remote job submission) toward end-user relevance.
- Since these solutions exist and others are already using them (and they're free), it's easier to reuse than to reinvent.
  - And compatibility with other Grid systems comes for free!

Continue Learning

- Visit the Globus Alliance website at: www.globus.org
- Read the book: *The Grid: Blueprint for a New Computing Infrastructure (2nd edition)*
- Talk to others who are using the Toolkit: discuss@globus.org (subscribe first)
- Participate in standards organizations: GGF, OASIS, W3C, IETF